

100

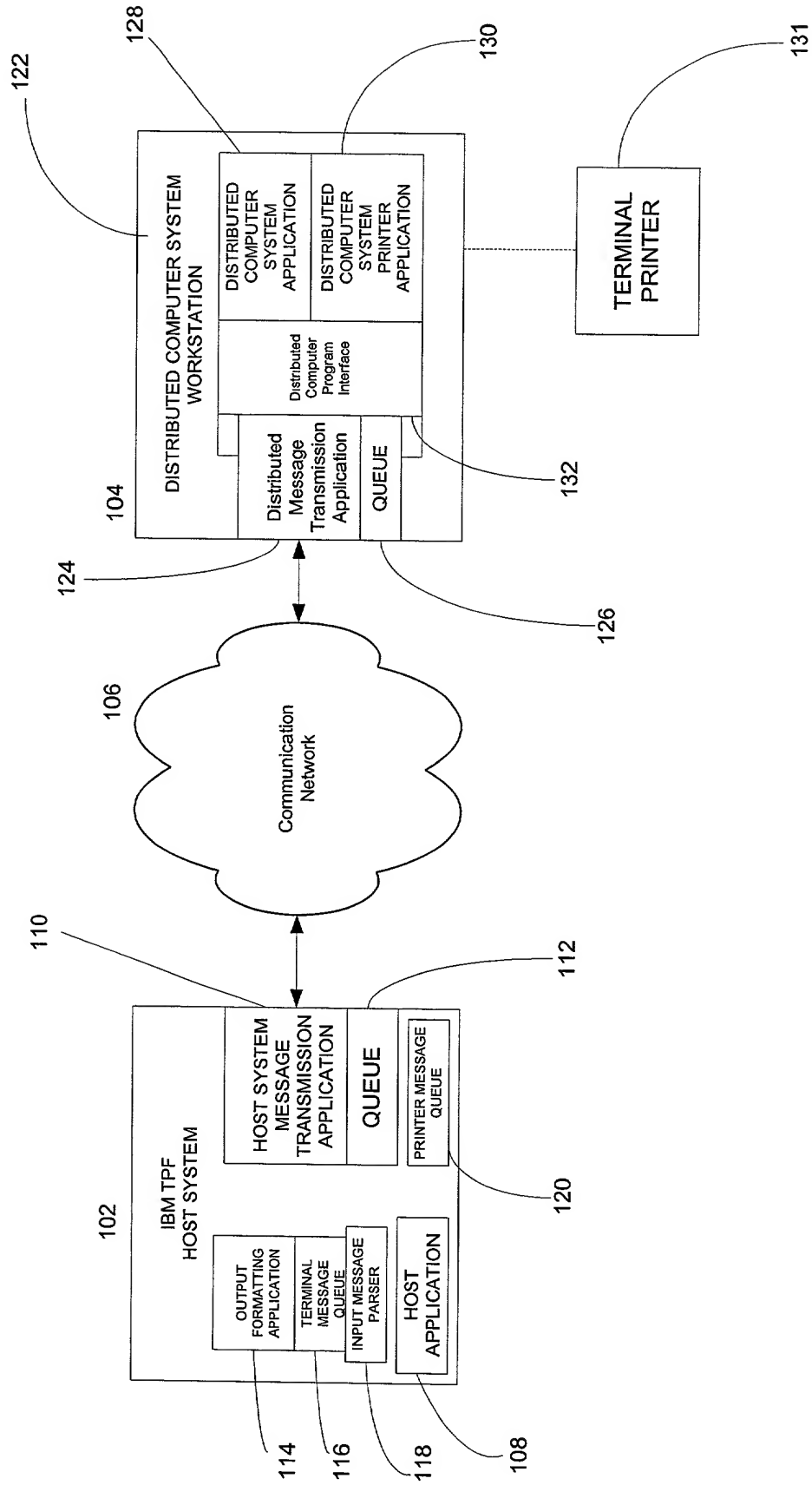
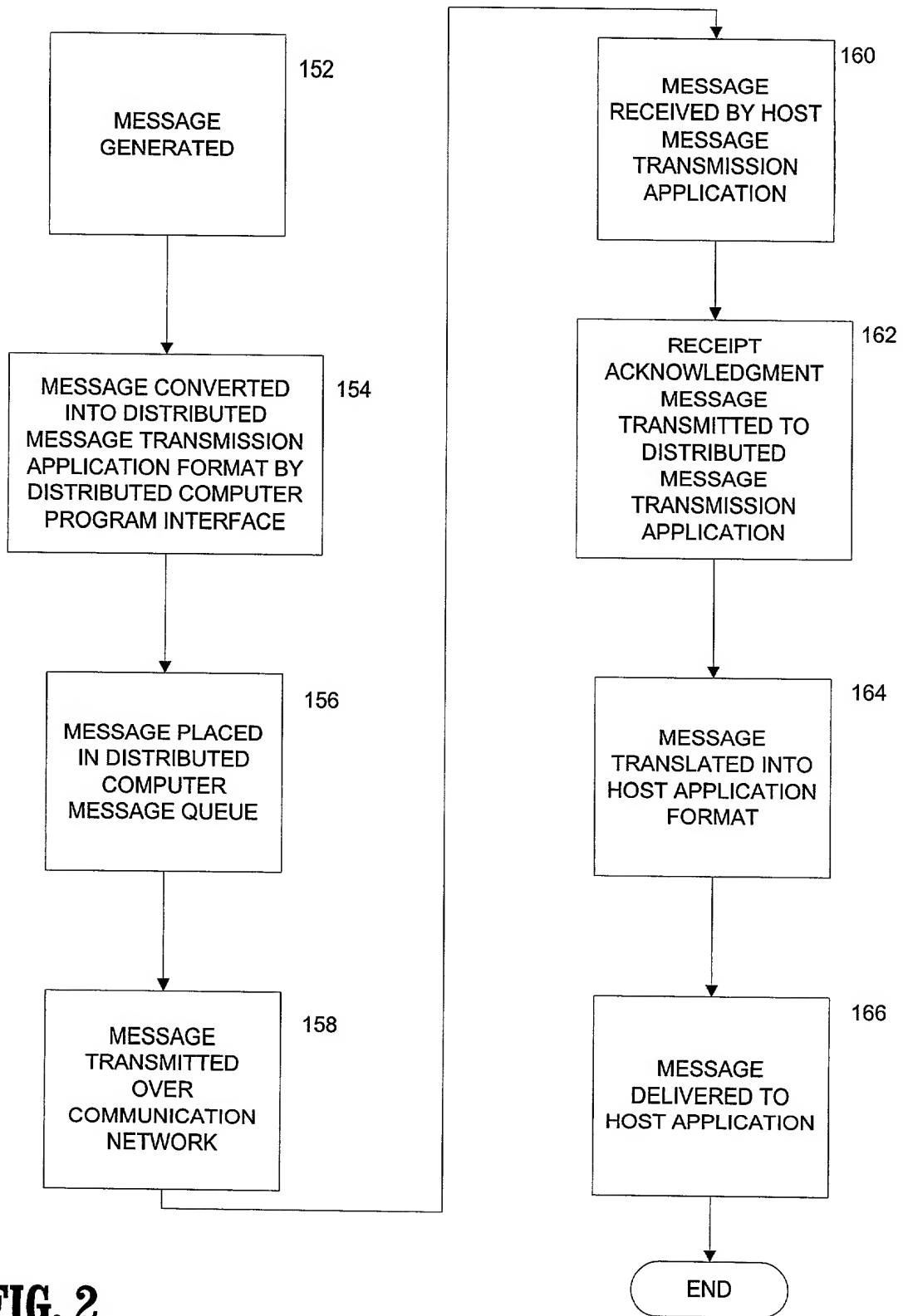


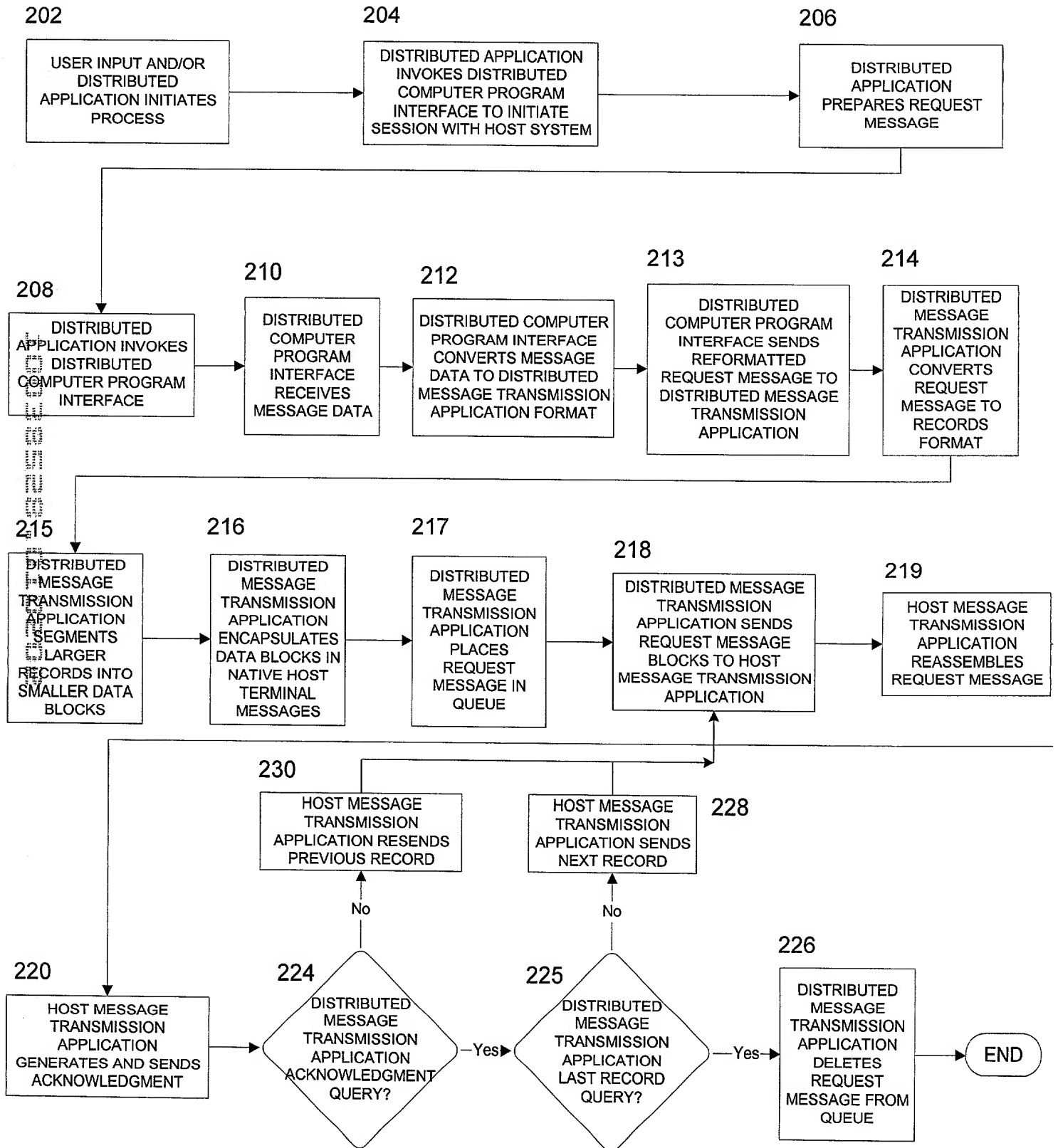
Fig. 1

150



**FIG. 2**

200



**FIG. 3**

FIG. 4A is a block diagram of a file 300. The file 300 is composed of a sequence of records, including Record 1, Record 2, Record 3, Record 4, and Record N. A bracket labeled 302a-n indicates the entire sequence of records from Record 1 to Record N.

300

File



302a-n

FIG. 4A

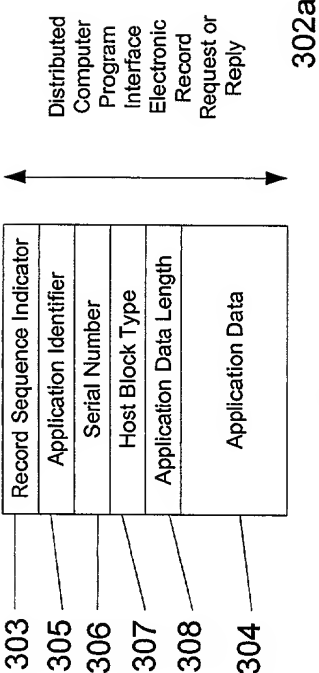


FIG. 4B

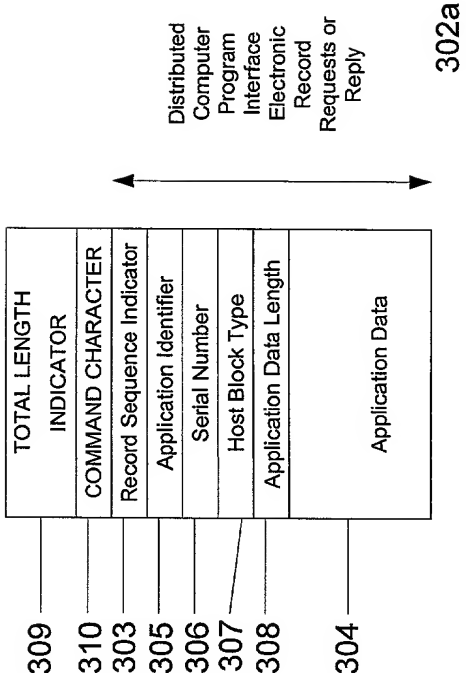
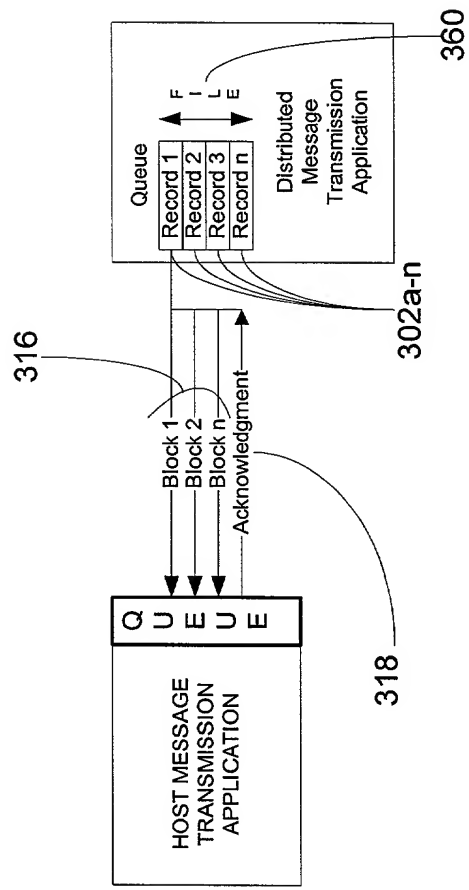
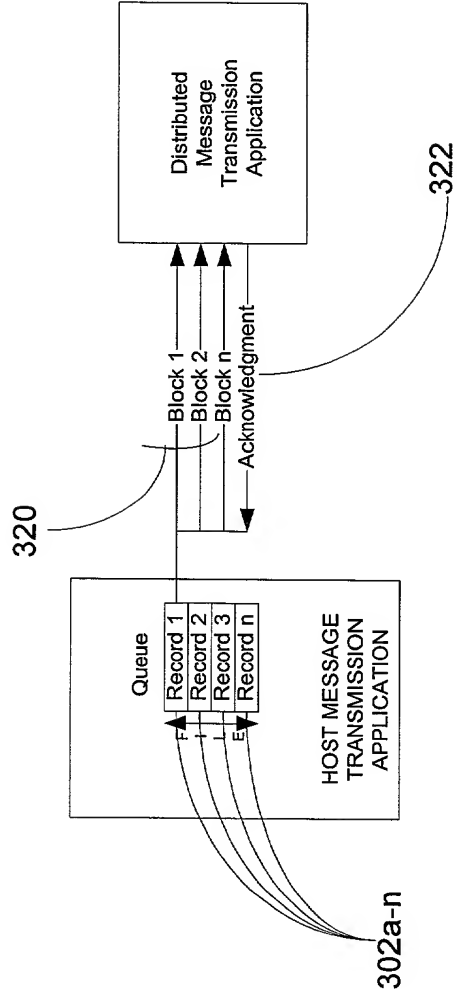


FIG. 4C



**FIG. 5**



**FIG. 6**



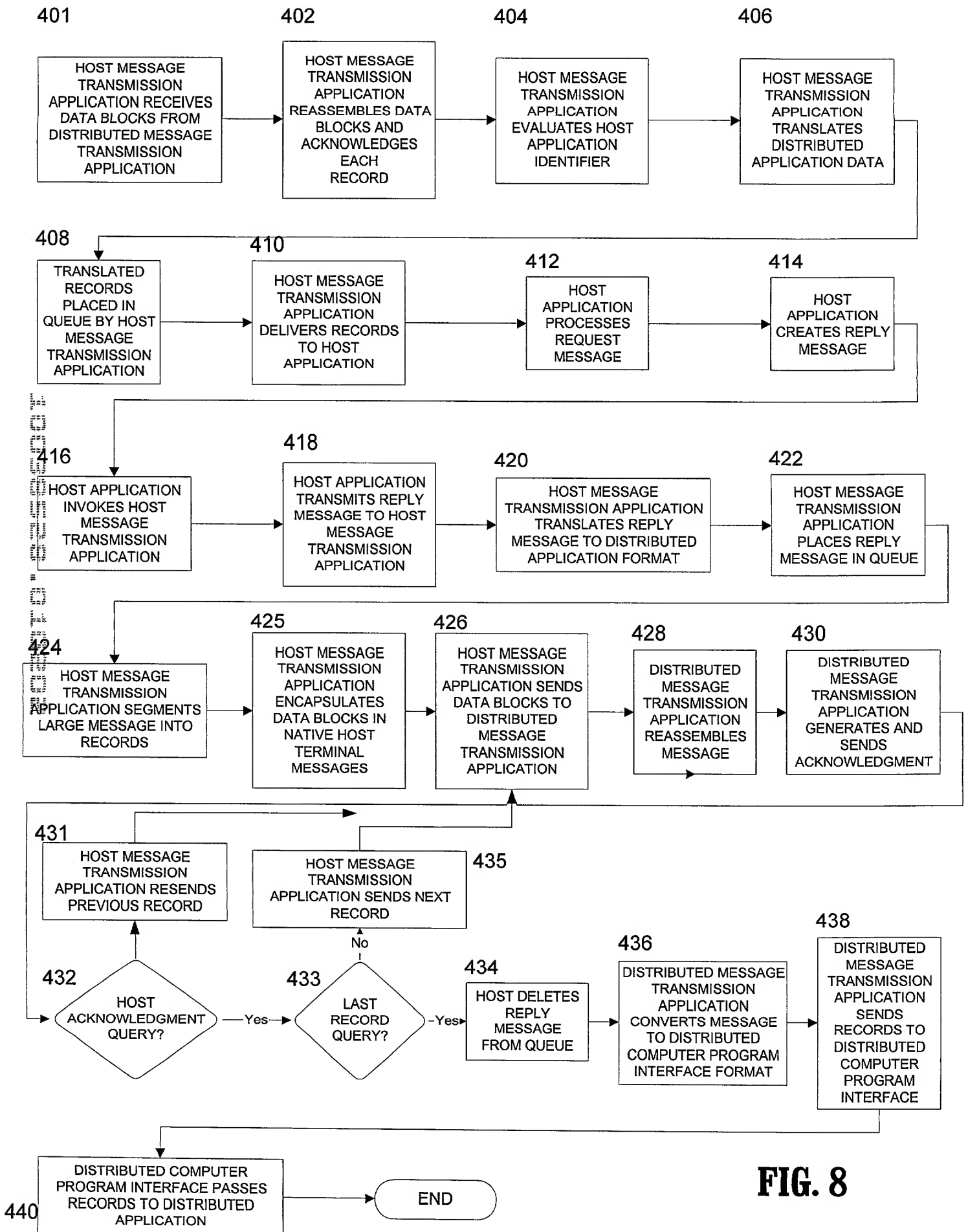


FIG. 8



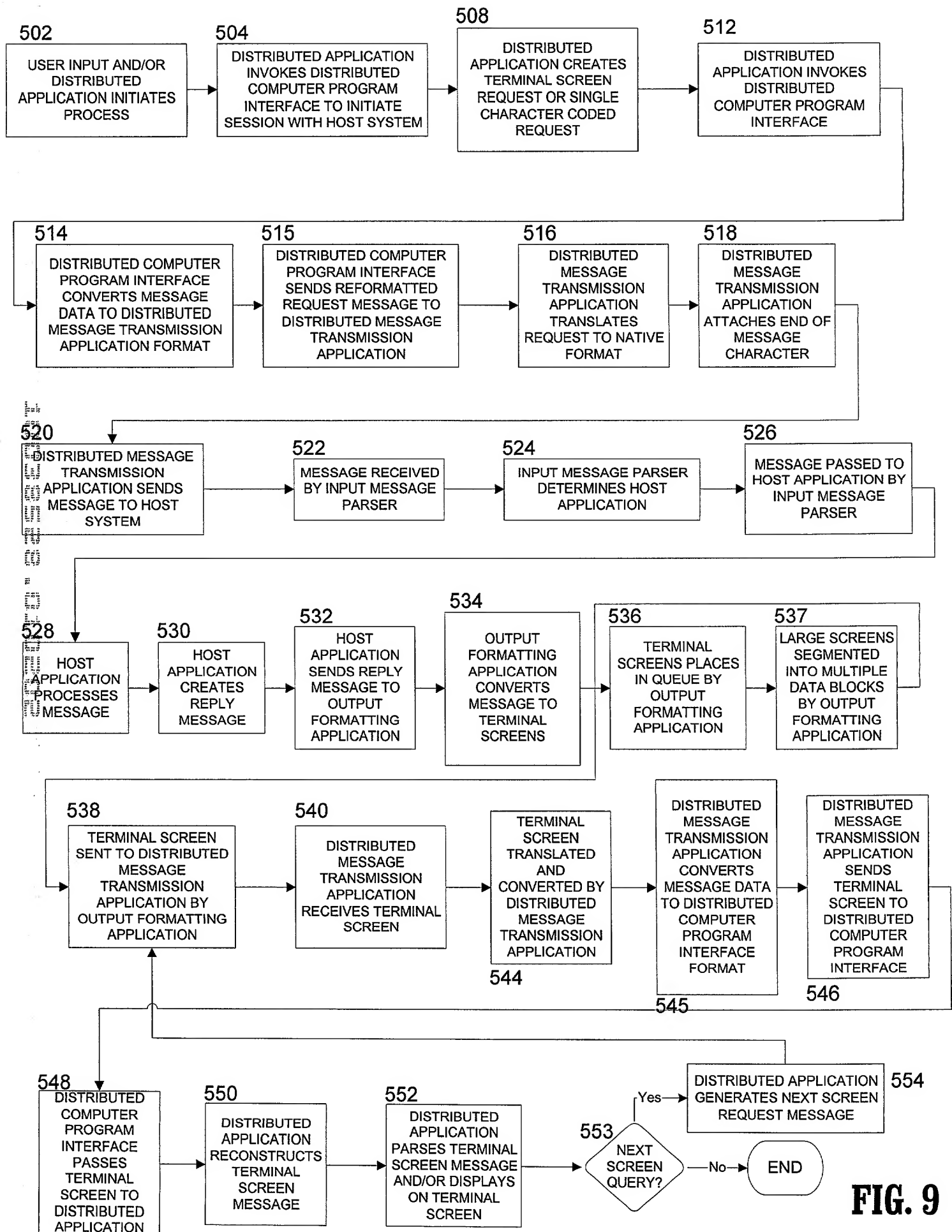
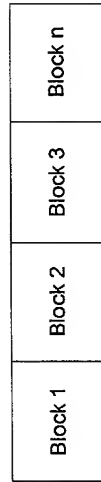


FIG. 9

FIG. 10 is a block diagram of a terminal screen reply message 600. The message 600 is composed of a plurality of blocks, including Block 1, Block 2, Block 3, and Block n. The blocks are arranged in a sequence, and the message 600 is labeled as 600. The blocks are connected by a curved line, and the label 602a-n is positioned below the blocks.

600

Terminal Screen Reply Message

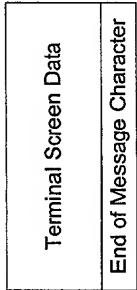


602a-n

FIG. 10

FIG. 11A is a block diagram of a Native Terminal Request Block 610. The block 610 is divided into two sections. The top section 612 is labeled "Terminal Screen Data". The bottom section 614 is labeled "End of Message Character".

610



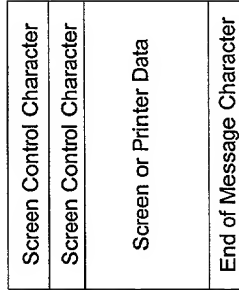
612

614

Native  
Terminal  
Request  
Block

**FIG. 11A**

620



626

628

622

624

Native  
Terminal  
or Printer  
Reply  
Block

**FIG. 11B**

IBM CORPORATION  
ARMONK, NEW YORK 10504  
U.S. PATENT OFFICE  
COMMUNICATIONS SECTION  
WASHINGTON, D.C. 20540

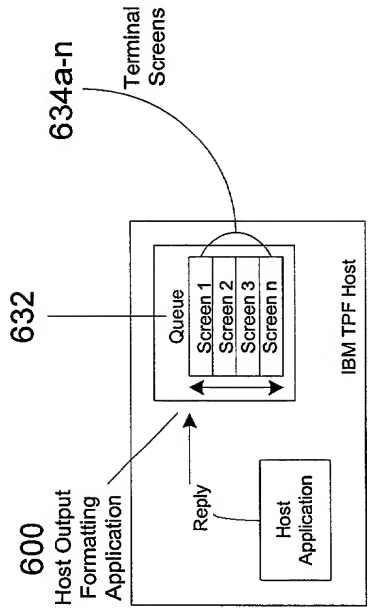
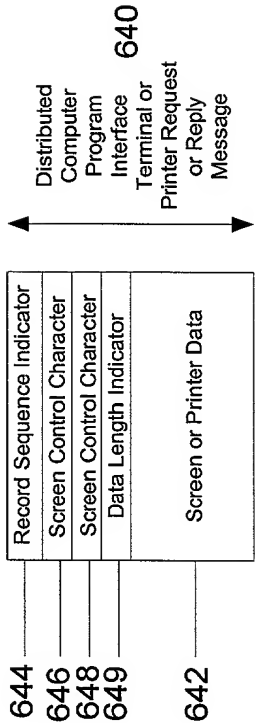
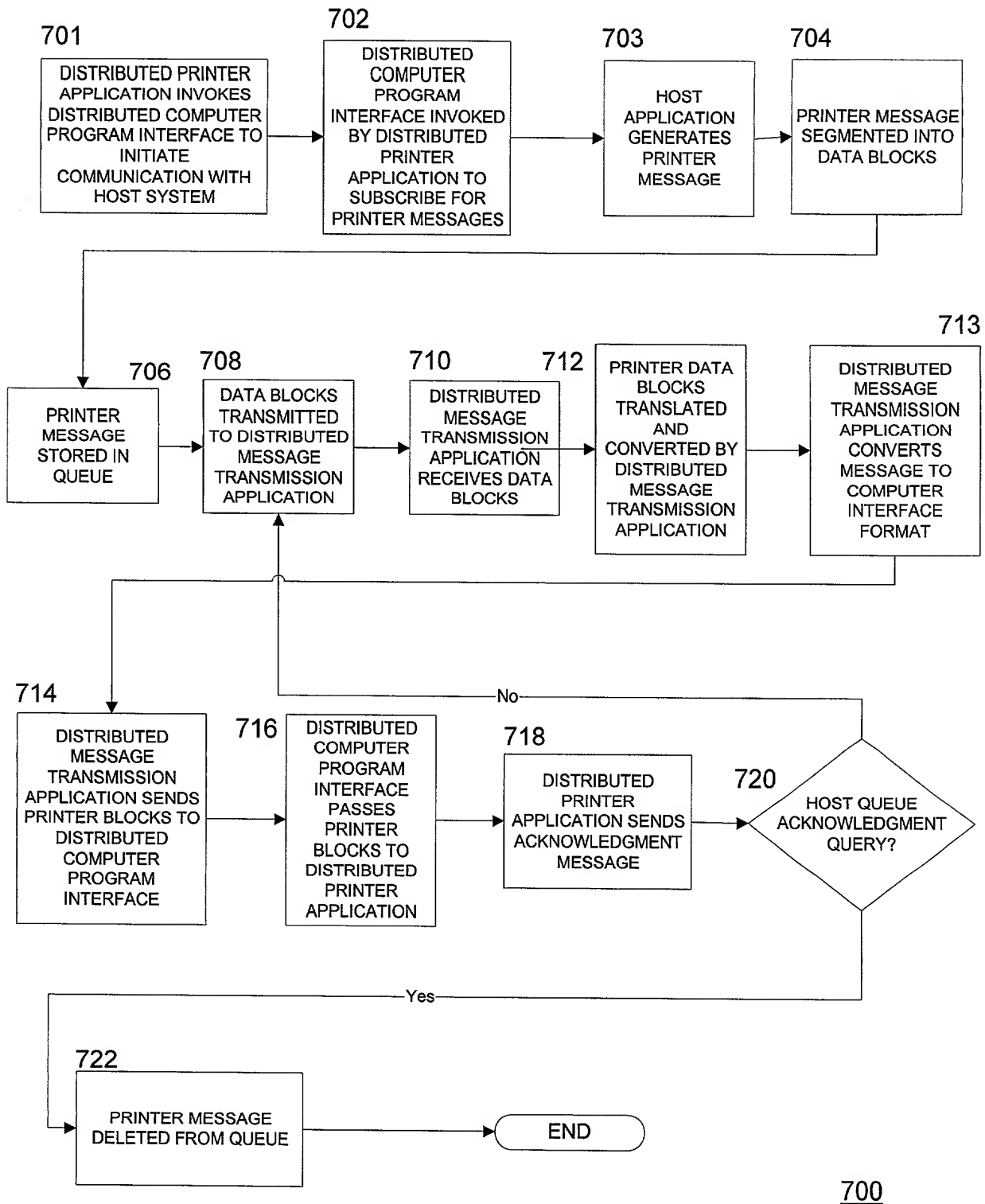


FIG. 12

FIG. 13 is a block diagram of a data structure for a distributed computer program interface terminal or printer request or reply message. The data structure is shown as a sequence of fields: Record Sequence Indicator (644), Screen Control Character (646), Screen Control Character (648), Data Length Indicator (649), and Screen or Printer Data (642).



**FIG. 13**

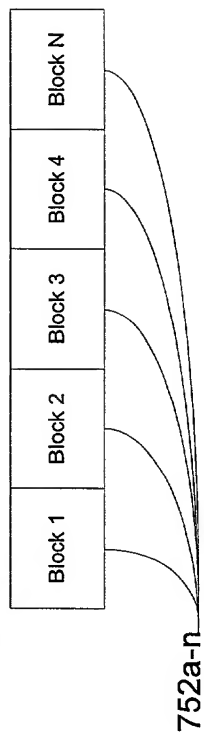


**FIG. 14**

[illegible]

750

Printer Reply Message



**FIG. 15**

755

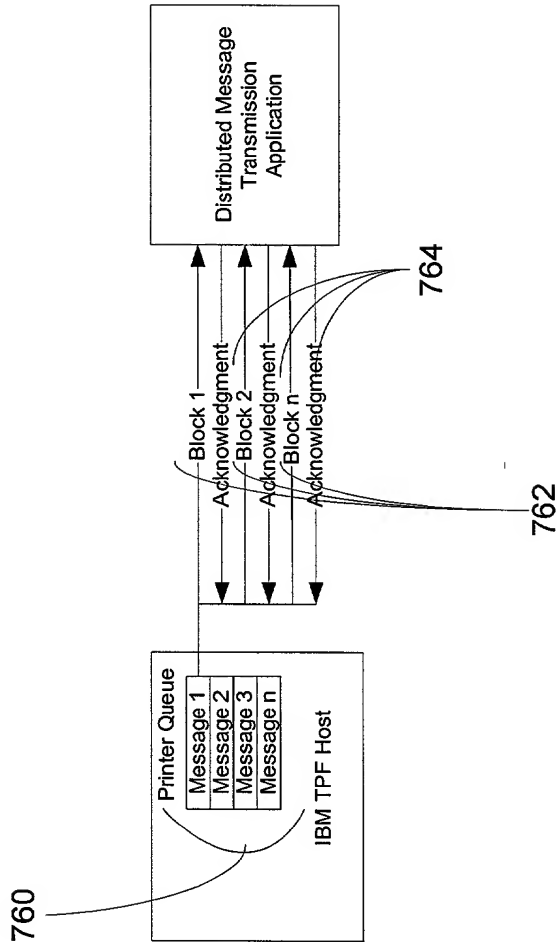
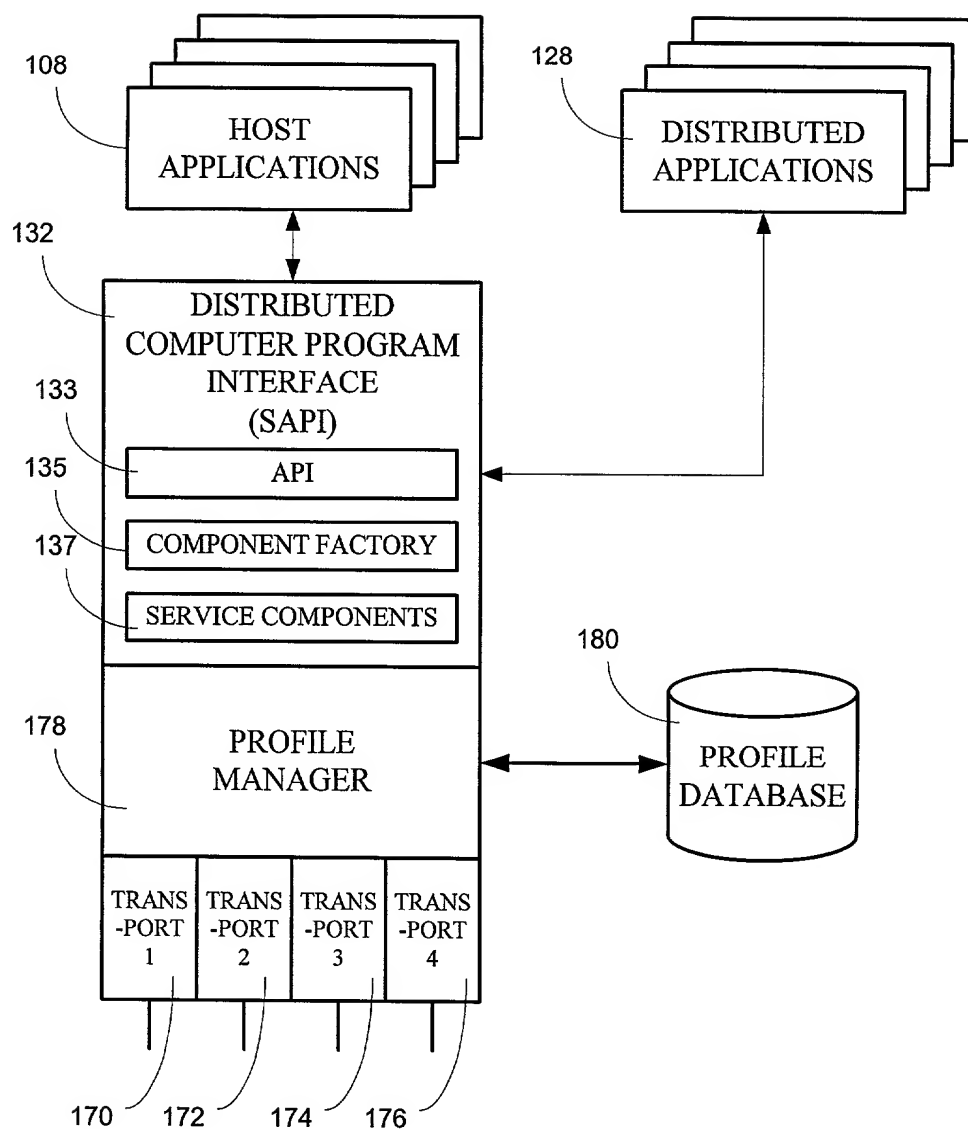
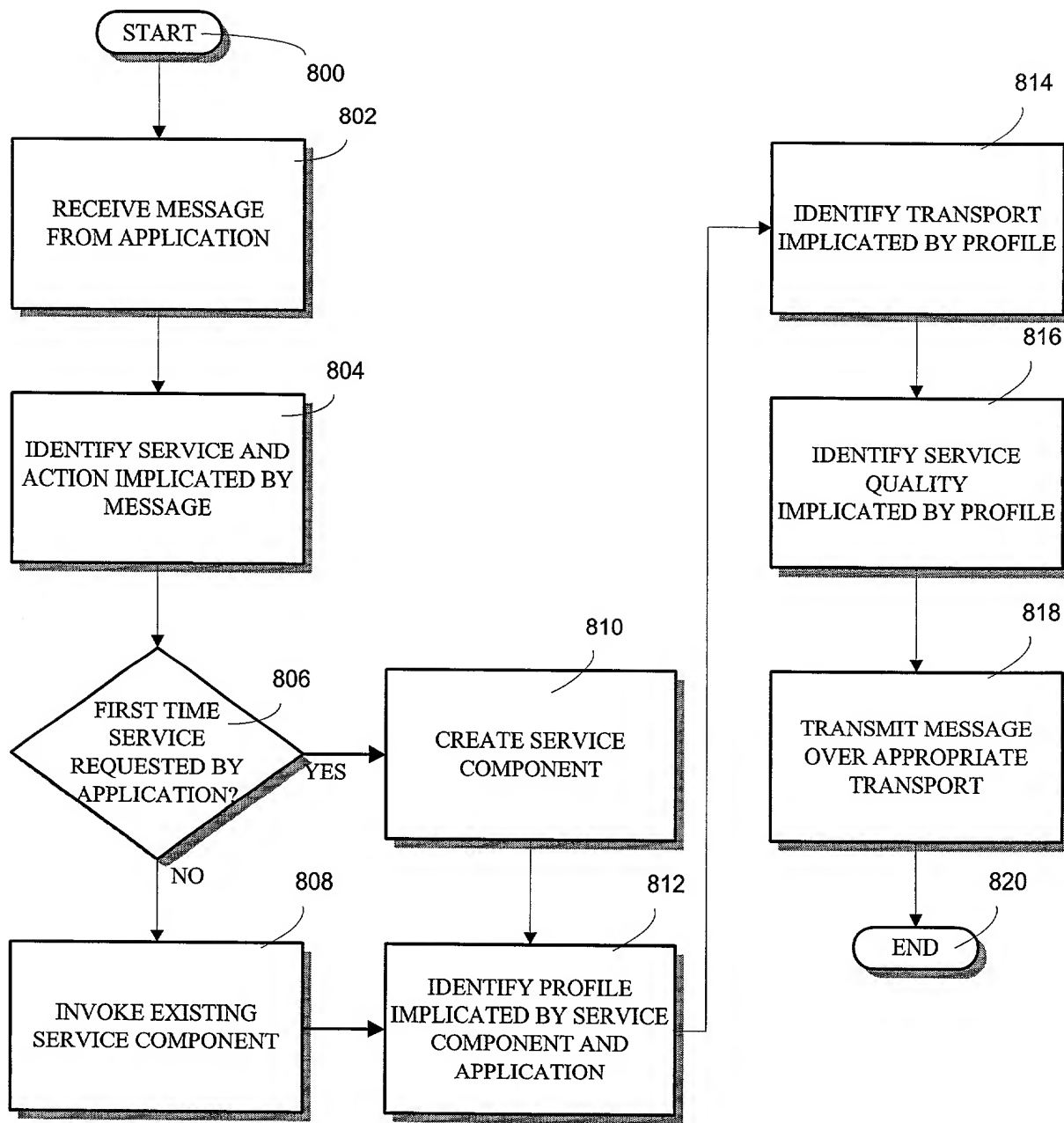


FIG. 16

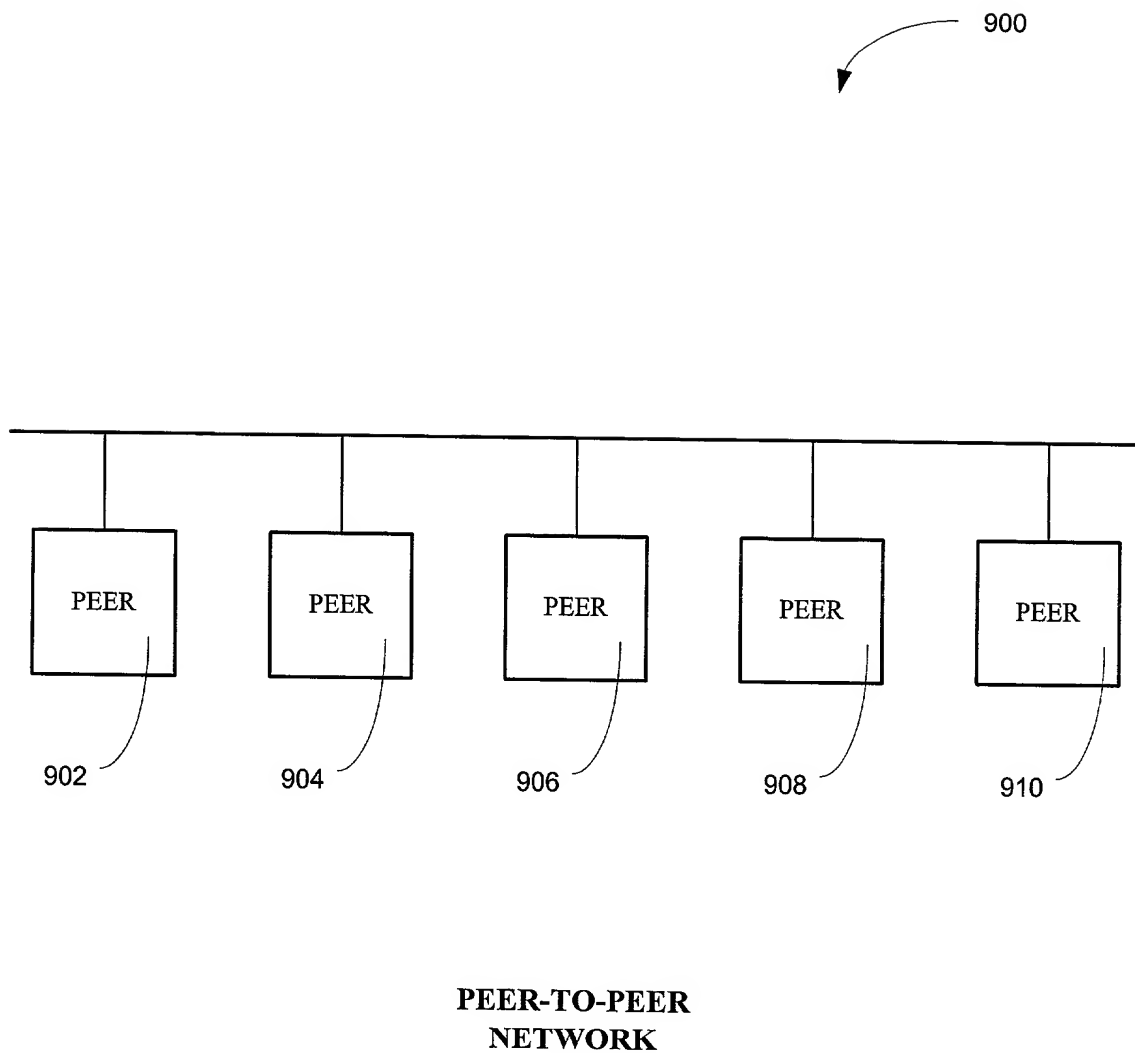




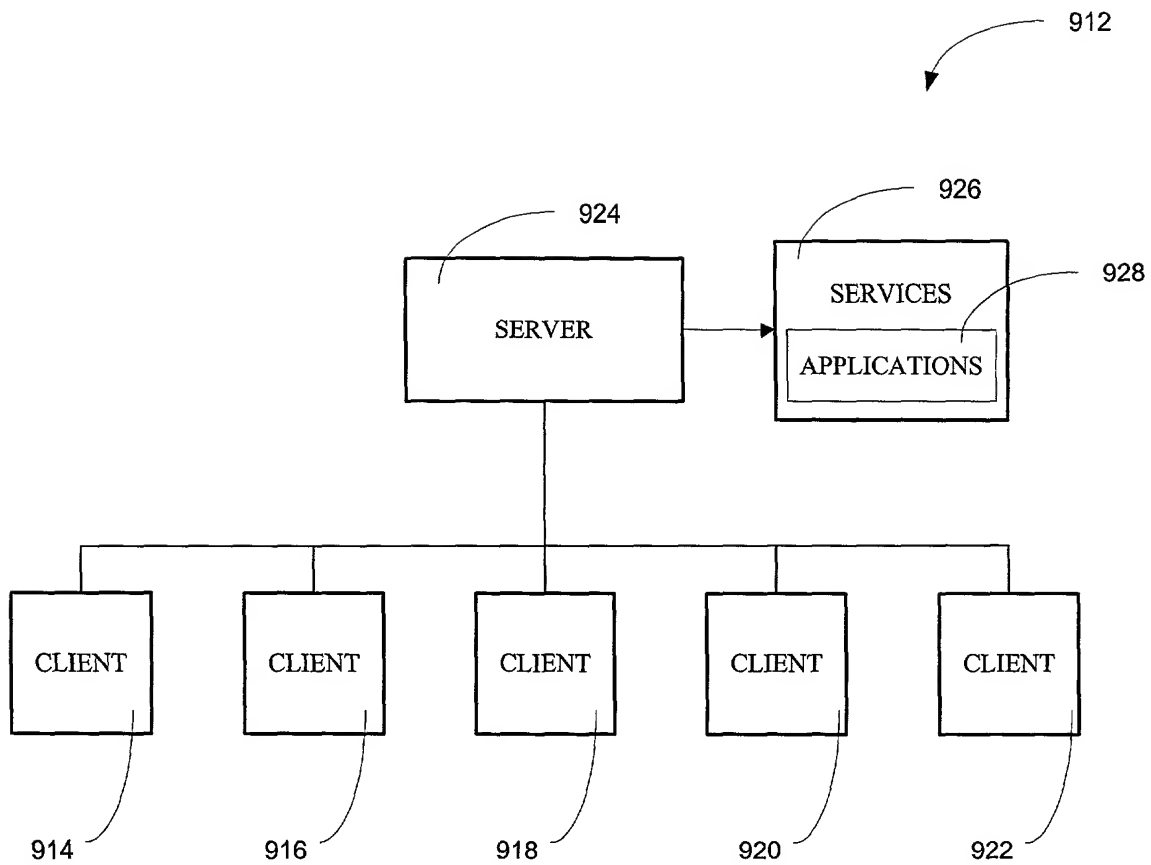
**Fig. 17**



**Fig. 18**

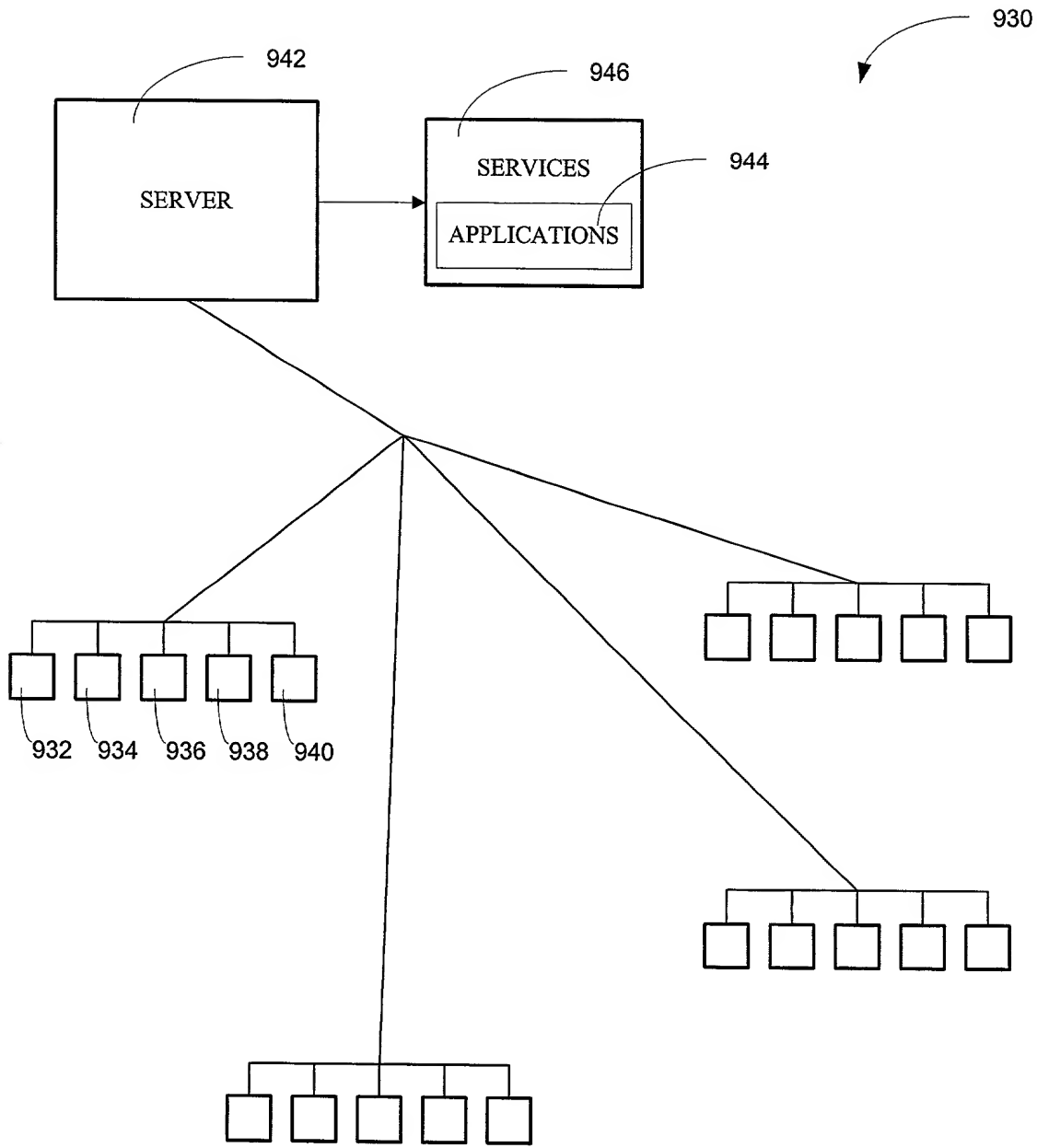


**Fig. 19**



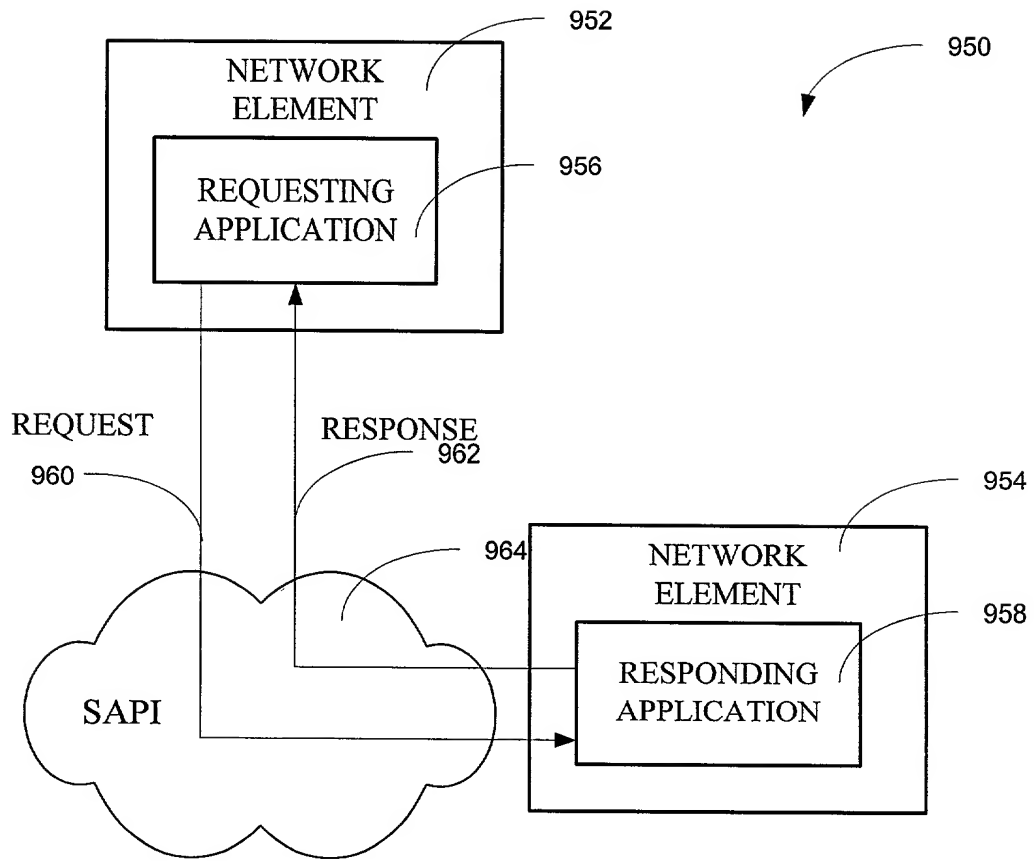
**CLIENT-SERVER  
NETWORK**

**Fig. 20**



WIDE AREA  
NETWORK

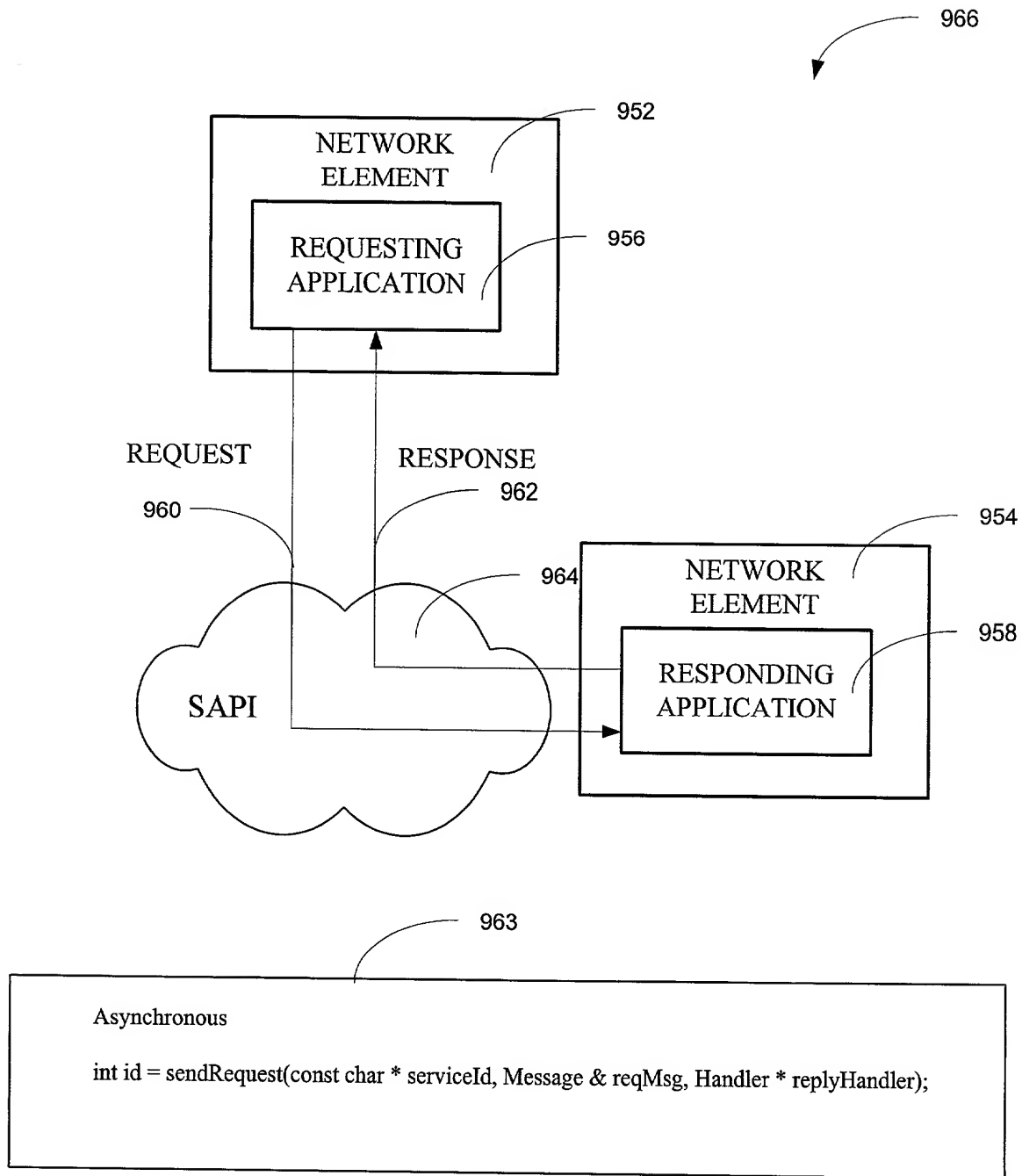
**Fig. 21**



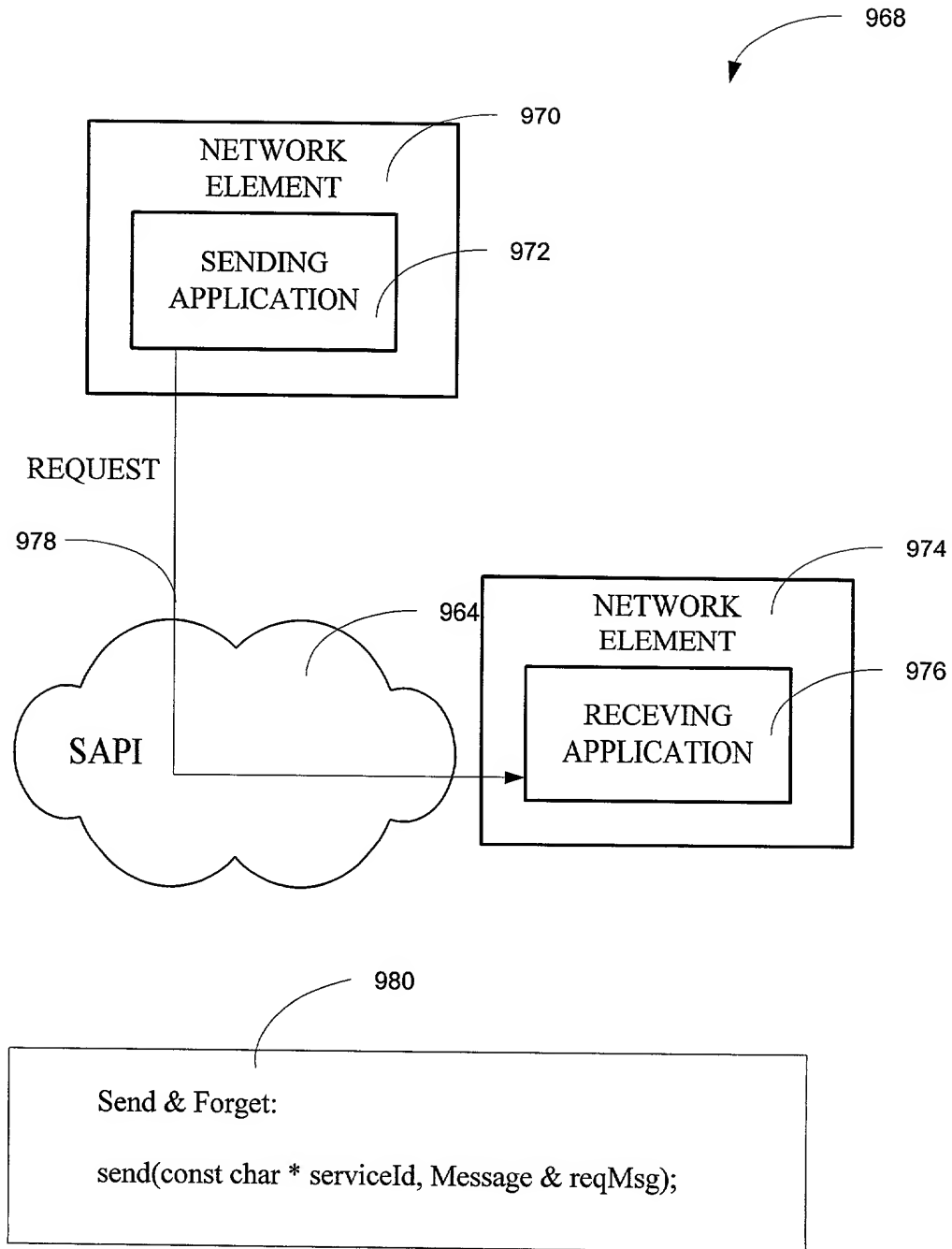
Synchronous mode:

Message \* reply = call(const char \* serviceId, Message & reqMsg);

**Fig. 22**

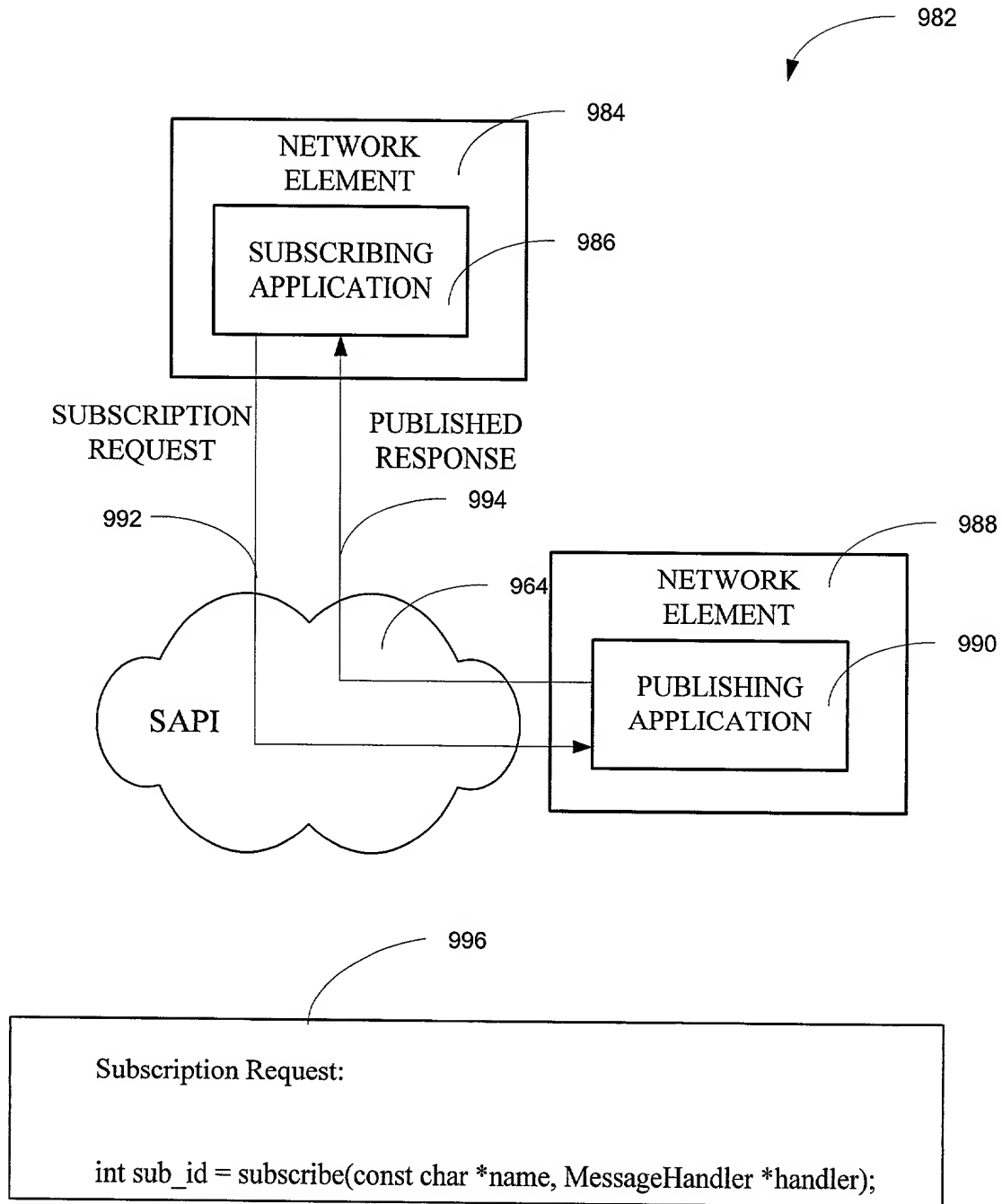


**Fig. 23**



**Fig. 24**





**Fig. 25**